SPARSPAK: Waterloo Sparse Matrix Package
User's Guide for SPARSPAK-B

Alan George
Esmond Ng

Department of Computer Science
University of Waterloo
Waterloo, Ontario, CANADA

Research Report CS-84-37

November 1984

Table of Contents

SPARSPAK: Waterloo Sparse Matrix Package

User's Guide for SPARSPAK-B

A collection of modules to be used with SPARSPAK-A for solving sparse constrained linear least squares problems

Alan George[†]

Esmond Ng[†]

Research Report CS-84-37

SPARSPAK-B User's Guide

This document describes the use of a system called SPARSPAK-B, which is an enhancement to the SPARSPAK-A package to allow for the solution of large sparse constrained (and unconstrained) linear least squares problems. SPARSPAK-B consists of new interface subroutines and some additional underlying subroutines, but makes extensive use of subroutines in the SPARSPAK-A package. SPARSPAK-B cannot run without SPARSPAK-A.

Although SPARSPAK-B depends upon SPARSPAK-A, it is in an important sense independent; a user can simultaneously solve a least squares problem and a problem from the class treated by the basic SPARSPAK-A package.

† Department of Computer Science, University of Waterloo, Waterloo, Ontario, CANADA.

IMPORTANT NOTE

The numerical algorithm used in *LSQSLV* (see Section 2.5) is a prototype implementation of an algorithm that is due to Bjorck [1]. The behaviour of the numerical algorithm, in the presence of roundoff errors in finite precision computer arithmetic, is not well-understood. In the course of solving a constrained linear least squares problem, some small dense subproblems may have to be solved. Some of these resulting subproblems may be sensitive to roundoff errors and numerical solutions to these subproblems may be inaccurate when a small tolerance is supplied to *LSQSLV*.

We would appreciate receiving any comments and feedback the user may have in using the package to solve practical problems. Such comments and feedback are important and useful since they may allow us to refine the numerical algorithm in the future.

When the package fails to produce an accurate solution, we would be grateful if the user could send us a copy of the data (if possible) so that we may locate where problems occur.

Please send comments and feedback to

Dr. Alan George or Dr. Esmond Ng
Department of Computer Science
University of Waterloo
Waterloo, Ontario
CANADA N2L 3G1

Telephone: 519-885-1211, ext 3473 (Dr. Alan George)
519-885-1211, ext 2517 (Dr. Esmond Ng)

1. Introduction and basic structure of SPARSPAK-B

SPARSPAK-B is designed to solve the problem

$$\min_{\beta \in \Omega} \| W_X(X\beta - y) \|_2 \quad ,$$

where

$$\Omega = \{ \beta \mid \beta \text{ minimizes } \| W_Y(Y\beta - z) \|_2 \} \quad .$$

Here X and Y are respectively $m \times n$ and $p \times n$ sparse matrices, y and z are vectors of length m and p respectively, and β is a vector of length n. The matrices W_X and W_Y are respectively $m \times m$ and $p \times p$ diagonal weight matrices. The package is capable of handling the very general problem in which X and Y may be either of full rank or rank-deficient, and there is no restriction on the dimensions of X and Y. Moreover, the constraints need not be consistent. When the constrained linear least squares problem does not have a unique solution, the package computes the solution which has minimal Euclidean norm.

Even though SPARSPAK-B is designed to solve a very general constrained linear least squares problem, it can also handle problems that may be much simpler than this general case. For example, the package is capable of solving an unconstrained linear least squares problem; that is, $p=0$. In particular, SPARSPAK-B can be used to solve a sparse general square system of linear equations ($p=0$ and $m=n$).

As we have noted above, the package does not impose any restrictions on the dimensions of X and Y. That is, the package also handles the case in which $m=0$ but $p \neq 0$ (even though such a problem may not have any physical meaning). In this situation, SPARSPAK-B simply treats the nonempty constraints as a linear least squares problem and computes the minimal norm least squares solution.

The basic computational technique used in SPARSPAK-B is due to George and Heath [3]. Let M denote the $(m+p) \times n$ matrix $\begin{bmatrix} Y \\ X \end{bmatrix}$. It is assumed that the Cholesky factor of M^TM is sparse. Then M is reduced to upper trapezoidal form $\begin{bmatrix} R \\ O \end{bmatrix}$ by applying Givens transformations and Gaussian eliminations to the rows of M. Here R is an $n \times n$ upper triangular matrix. Finally the upper triangular matrix R is used to compute the solution. It can be shown that the structure of R is contained in the structure of the Cholesky factor of the symmetric positive definite matrix M^TM. Thus, using techniques developed for solving sparse symmetric positive definite systems, one can *predict* the structure of R by analyzing the structure of M^TM [7]. This allows a storage scheme for R to be set up before numerical factorization begins and hence the numerical computation of R can be carried out using a static data structure. Experience has shown that this approach is efficient (both in terms of storage and execution time) compared to schemes that employ dynamic storage allocation [4].

In most cases, the matrix M^TM is sparse when both the "least squares matrix" X and the "constraint matrix" Y are sparse. However, there are instances in which M^TM is dense even though X and Y are sparse. Fortunately this usually occurs when there are only a few "dense rows" in X and Y. Such problems can be handled in a special way by the package. More precisely, the least squares problem and the constraints may be regarded as being partitioned as follows:

$$X\beta - y = \begin{bmatrix} A \\ B \end{bmatrix}\beta - \begin{bmatrix} y_A \\ y_B \end{bmatrix}$$

$$Y\beta - z = \begin{bmatrix} E \\ F \end{bmatrix}\beta - \begin{bmatrix} z_E \\ z_F \end{bmatrix}$$

where A and E contain respectively the sparse equations and constraints, and B and F contain respectively the dense equations and constraints. It is now assumed that M denotes the matrix $\begin{bmatrix} E \\ A \end{bmatrix}$. Furthermore, it is assumed that the Cholesky factor of M^TM is sparse. The package reduces M to upper trapezoidal form using the approach described above. Then the upper trapezoidal form, together with B and F, are used to derive the solution to the original problem. The algorithm is due to Bjorck [1]. Detailed description of the implementation can be found in [11].

In general the algorithms, the data structures and the storage management for solving sparse matrix problems are quite complicated. Thus, in order to insulate the user from these considerations, a set of simple user interface subroutines is used in the package. These interface subroutines will be described in detail in the following sections.

The user and the package interact to solve the problem through the following basic steps.

Step 1. The user supplies the rows of $[X \mid y]$ and $[Y \mid z]$ to the package, in any order, along with the corresponding diagonal element of W_X and W_Y.

Step 2. The user calls a subroutine which initiates a reordering of the columns of $\begin{bmatrix} E \\ A \end{bmatrix}$, in order to preserve sparsity in subsequent calculations. (See Section 2.3.)

Step 3. The user calls a subroutine which initiates a reordering of the rows of X and Y according to one of several criteria. (See Section 2.4.)

Step 4. The user calls a subroutine which computes the solution β.

The package has facilities to allow the user to compute conveniently the norm of the residual vectors

$$|| y - X\beta ||_2 \qquad \text{and} \qquad || z - Y\beta ||_2 \quad ,$$

where β is the computed solution.

Important notes:

1. Note that SPARSPAK-B is designed to handle problems in which X and Y may have any dimensions. However, the package will perform more efficiently (in terms of storage and execution time) when $m+p \geq n$ (that is, overdetermined problems). When $m+p \ll n$ (that is, underdetermined problems), the package will still be able to solve the problems, but both the storage and time requirements may be high. See [5] for some examples and algorithms for handling sparse unconstrained underdetermined problems.
2. The discussions above indicate that the package employs heavily techniques for handling sparse symmetric positive definite systems of linear equations. Indeed, SPARSPAK-B makes extensive use of the subroutines in SPARSPAK-A, which is a package designed for solving efficiently sparse symmetric positive definite linear systems [2]. SPARSPAK-B cannot run

without SPARSPAK-A. Although SPARSPAK-B depends upon SPARSPAK-A, it is in an important sense independent. A user can simultaneously solve a sparse constrained linear least squares problem and a sparse symmetric positive definite linear system in the same program.

An early version of SPARSPAK-B was designed and implemented by Dr. M.T. Heath at the Oak Ridge National Laboratory. This early version includes algorithms for solving the basic sparse unconstrained linear least squares problems, and for handling constraints, dense rows and rank deficiency [12]. Those algorithms are special cases of a more general algorithm due to Bjorck for handling more general sparse constrained linear least squares problems [1]. This general algorithm is used in the current version of SPARSPAK-B. The design of SPARSPAK-B is similar to that of SPARSPAK-A [2]. The reader is referred to [6] for a discussion of the design and implementation issues.

2. Modules of SPARSPAK-B and how to use them

2.1. User mainline program and an example

SPARSPAK-B allocates all of its storage from a single one-dimensional floating-point array[1] which for purposes of discussion we will denote by T. In addition, the user must provide its size *MAXSB*, which is transmitted to the package via a common block *SPBUSR*, (SPARSPAK−B USER), which has eight variables.

```
      COMMON /SPBUSR/ MSGLVB, IERRB, MAXSB, NCOLS,
                      NSEQNS, NDEQNS, NSCONS, NDCONS
```

Here *MSGLVB* is the message level indicator which is used to control the amount of information printed by the package. The second variable *IERRB* is an error code, which the user can examine in his mainline program for possible errors detected by the package. Detailed discussion of the roles of *MSGLVB* and *IERRB* is provided in Section 6. The variable *NCOLS* is the number of columns in X and Y, and *NSEQNS*, *NDEQNS*, *NSCONS*, *NDCONS* are respectively the number of rows in A, B, E and F. Thus, $NSEQNS+NDEQNS=m$ and $NSCONS+NDCONS=p$.

The following program illustrates how one might use SPARSPAK-B. The various subroutines referenced are described in the subsequent parts of this section. The problem solved is assumed to be stored on an external file (FORTRAN unit 1) in a binary format.

```
      INTEGER  SUBS(10)
      INTEGER  FILE, IERRB, INPUT, K, MAXSB, MSGLVB,
     1         NCOLS, NDCONS, NDEQNS, NSCONS, NSEQNS,
     1         NSUBS, OPTION, ROWNUM, TYPE, TYPTOL
      REAL  T(5000), VALUES(10)
      REAL  RESCON, RESEQN, RHS, TOL, WEIGHT
      COMMON /SPBUSR/ MSGLVB, IERRB, MAXSB, NCOLS,
     1                NSEQNS, NDEQNS, NSCONS, NDCONS
C
C     ---------------
C     INITIALIZATION.
C     ---------------
      INPUT = 1
      FILE = 2
      OPTION = 0
      TOL = 1.0E-6
      TYPTOL = 0
C
      CALL  SPRSPK
      MSGLVB = 2
      MAXSB = 5000
C
      CALL  FILEB ( FILE )
C     ------------------------------
C     INPUT ROWS FROM EXTERNAL FILE.
C     ------------------------------
      REWIND  INPUT
  100 CONTINUE
          READ (INPUT)  ROWNUM, TYPE, NSUBS,
     1                  (SUBS(K),VALUES(K),K=1,NSUBS),
```

(1) Declared either REAL or DOUBLE PRECISION, depending on the version of SPARSPAK-A and SPARSPAK-B that is available. The examples in this manual assume a single precision version is being used.

```
     1                        RHS, WEIGHT
            IF  ( NSUBS .EQ. 0 )  GO TO 200
            CALL  INXYWB ( ROWNUM, TYPE, NSUBS, SUBS, VALUES,
     1                     RHS, WEIGHT, T )
            GO TO 100
  200     CONTINUE
C           ----------------------
C           ORDER COLUMNS AND ROWS.
C           ----------------------
            CALL  ORCOLB ( T )
            CALL  ORROWB ( OPTION, T )
C           ------------------------------------------
C           COMPUTE LEAST SQUARES SOLUTION AND RESIDUALS.
C           ------------------------------------------
            CALL  LSQSLV ( TOL, TYPTOL, T )
            CALL  RESIDB ( RESEQN, RESCON, T )
C           ------------------------------------------
C           PRINT THE SOLUTION, FOUND IN THE FIRST NCOLS
C           LOCATIONS IN THE WORKING STORAGE ARRAY T, AND
C           PRINT THE RESIDUALS.
C           ------------------------------------------
            WRITE (6,11)  (T(K),K=1,NCOLS)
   11       FORMAT ( / 10H SOLUTION / (5F12.5) )
            WRITE (6,22)  RESEQN, RESCON
   22       FORMAT ( / 22H EQUATION RESIDUAL    =, F12.5
     1              / 22H CONSTRAINT RESIDUAL =, F12.5 )
C           ---------------------------------------
C           PRINT STATISTICS GATHERED BY SPARSPAK-B.
C           ---------------------------------------
            CALL  STATSB
C
            STOP
          END
```

Note: If the SPARSPAK-B package available to you is a double precision version, the REAL declarations in this example should be changed to DOUBLE PRECISION.

The module *SPRSPK* must be called before any part of the package is used. Its role is to initialize some system parameters (e.g. the logical unit numbers for output files), to set default values for options (e.g. the message level indicator), and to initialize the timing routine. The routine needs only to be called once in the user program, and the FORTRAN statement is shown below.

CALL SPRSPK

(In fact, *SPRSPK* is the initialization routine for the entire SPARSPAK package, including both SPARSPAK-A and SPARSPAK-B.) Note that the only variable in the common block *SPBUSR* that *must* be explicitly assigned a value by the user is *MAXSB*, although he may wish to set others as well, such as *MSGLVB*.

It is assumed that the subroutines which comprise SPARSPAK-A and SPARSPAK-B have been compiled into a *library*, and that the user can reference them from his FORTRAN program just as he references the standard FORTRAN library subroutines, such as *SIN*, *COS*, etc. Normally, a user will use only a small fraction of the subroutines provided in SPARSPAK-A and SPARSPAK-B.

SPARSPAK-B requires an external sequential file for the storage of intermediate results. Before beginning to solve a problem, the user must call a subroutine *FILEB* to tell SPARSPAK-B which FORTRAN unit it should use. The FORTRAN statement to be used is

CALL FILEB (IONUM)

where *IONUM* is the required FORTRAN logical unit.

Important Notes:

(1) The module *FILEB* must be called when a new problem is to be solved.

(2) The user is responsible for defining the external file for the FORTRAN logical unit *IONUM* using the appropriate system control statement or command. (This depends on the environment in which the program is being executed.) Furthermore, the file should be preserved throughout the execution of the program.

Warning:

The modules of SPARSPAK-B communicate with each other through labelled common blocks whose names are *SPKSYS*, *SPBUSR*, *SPBCON*, *SPBMAP*, and *SPBDTA*. (Note that *SPKSYS* is shared by SPARSPAK-A and SPARSPAK-B.) Thus, the user must not use labelled common blocks with these names in his program.

If these common block names cause conflicts in your program or at your computer installation, it is possible to have the package distributed with these common blocks having *specifically requested names*. These names should be specified when the package is acquired.

2.2. Modules for input of the problem

The subroutine *INXYWB* allows the user to provide the structure and the numerical values of X and Y to the package, along with the numerical values of y, z, W_X, and W_Y. They are provided one row at a time, as shown in the example below.

CALL INXYWB (ROWNUM, TYPE, NSUBS, SUBS, VALUES, RHS, WEIGHT, T)

The parameters of the subroutine are as follows.

ROWNUM: An integer variable associated with each row. Usually, $1 \leq ROWNUM \leq (m+p)$ and the rows of X and Y are labelled from 1 to $m+p$. The rows of X and Y can be intermixed in this labelling. It should be emphasized that *ROWNUM* is used only as a label for a row. In some contexts, it may be reasonable for several rows to have the same value for *ROWNUM*. The use of *ROWNUM* along with the use of *ORROWB* with *OPTION* set to 5 (see Section 2.4) allows the user to impose a specific row ordering.

TYPE: An integer variable having value 1, 2, 3 or 4, indicating the following.

1 - a row of A (sparse equation in X), with corresponding element of y and W_X.

2 - a row of B (dense equation in X), with corresponding element of y and W_X. (See below for an explanation.)

3 - a row of E (sparse constraint in Y), with corresponding element of z and W_Y.

4 - a row of F (dense constraint in Y), with corresponding element of z and W_Y. (See below for an explanation.)

NSUBS: An integer variable containing the number of nonzeros in the input row.

SUBS: An integer array containing the column indices (subscripts) of the nonzeros in the input row.

VALUES: A floating-point array containing the numerical values of the nonzeros in the input row in positions corresponding to the column indices stored in *SUBS*.

RHS: The right-hand side element of y or z corresponding to the input row.

WEIGHT: The diagonal element of the weight matrix W_X or W_Y corresponding to the input row. It is assumed that *WEIGHT* is positive.

T: The floating-point working storage array from which all storage for the package is allocated. (See Section 2.1 and the example there.)

Dense rows

Some problems may yield a few rows of X and of Y that have relatively many nonzeros. Such rows wreak havoc with the sparsity preservation techniques used in SPARSPAK-B. At the moment the package has no robust scheme for *deciding* which rows will cause unacceptable damage, but it does have a way of circumventing problems caused by such rows. Accordingly, the user can indicate dense rows of X or of Y by setting the corresponding value of the input parameter *TYPE* to 2 or 4 when such a row is input. See Example 2 in Section 8 for an illustration of the use of this feature.

Important Note:

The floating-point values transmitted to SPARSPAK-B by *INXYWB* are either *single or double precision floating-point numbers*, depending on the version of SPARSPAK-B being used. The examples in this manual assume that a single precision version of the package is being used.

2.3. Module for ordering the columns

Recall that the Cholesky factor of the symmetric positive definite matrix M^TM plays an important role in the solution process, where $M = \begin{bmatrix} E \\ A \end{bmatrix}$. It is well known that if M^TM is sparse, the sparsity of its Cholesky factor depends crucially on the symmetric ordering of the rows and columns of M^TM [7]. Note that a symmetric ordering of M^TM is the same as a column ordering of M. Thus, in order to reduce storage and execution time, one should find a "good" column ordering for M before any numerical computation begins so that the Cholesky factor of the reordered matrix is (hopefully) sparse. This can be achieved after the problem has been supplied to

the package using *INXYWB*. The column ordering process is invoked by executing the following FORTRAN statement.

CALL ORCOLB (T)

The algorithm used in SPARSPAK-B is an implementation of the minimum degree algorithm due to Liu [14]. The module *ORCOLB* is also responsible for setting up the appropriate data structures for the matrices involved in subsequent numerical computations.

Common Errors:

The most common cause of premature termination of the *ORCOLB* module is insufficient working storage. As mentioned above, this module performs two functions: *column ordering* and *storage allocation*. The ordering step determines the column permutation, and the allocation step sets up the appropriate data structures.

In general, the ordering and allocation subroutines require different amounts of storage. Furthermore, their storage requirements are often unpredictable, because the number of data structure pointers, and the number of nonzeros to be stored are not known until the subroutines have been executed.

Thus, the interface module *ORCOLB* may terminate in several distinctly different ways.

(a) There was not enough storage to execute the column ordering subroutine.

(b) The ordering was successfully obtained, but there was insufficient storage to initiate execution of the data structure set-up (storage allocation) subroutine.

(c) The data structure set-up subroutine was executed, and the amount of storage required for the data structure pointers etc. was determined, but there was insufficient storage for these pointers.

(d) The data structure was successfully generated, but there is insufficient storage for the actual numerical values in the upper trapezoidal matrix, so the next step (numerical computation) cannot be executed.

(e) *ORCOLB* was successfully executed, and there is sufficient storage to proceed to the next step.

If any of the above conditions occurs, the user may execute *SAVEB*, and re-initiate the computation after adjusting the storage declarations (either up or down) and executing RSTRTB[(2)]. If (a) or (b) occurs, information is supplied indicating the minimum value of *MAXSB* needed so that (c) or (d) will occur upon re-execution. If (c) occurs, the minimum value of *MAXSB* needed for (d) is provided.

When (c) or (d) occurs, after executing *SAVEB*, adjusting the storage declaration, then executing *RSTRTB*, one must again call *ORCOLB*. However, the interface will detect that the ordering and/or storage allocation have already been performed, and will skip that part of the computation.

(2) See Section 4 for details on how to use *SAVEB* and *RSTRTB*, and Examples 3 and 4 in Section 8.

2.4. Module for reordering the rows

The execution time or the numerical stability of the module *LSQSLV* described in Section 2.5 can be affected significantly by the row ordering [3,13]. Accordingly, SPARSPAK-B provides a row-ordering module which may be invoked (optionally) by the following FORTRAN statement.

CALL ORROWB (OPTION, T)

When the weights W_X and W_Y vary widely in magnitude, it is important for numerical accuracy that the rows of A and E be arranged in order of increasing weight. See [13] for details. This can be achieved by setting the integer parameter *OPTION* to 1.

If the rows of $W_X X$ and $W_Y Y$ do not vary greatly, the user may wish to sort the rows in order to reduce execution time, although the user should be aware that this sorting might require *more* storage than would otherwise be required to solve the problem. The following options are provided.

OPTION	Details
0	Rows are processed in the order they were supplied, regardless of the values of *ROWNUM* (see Section 2.2).
1	Rows are sorted in order of increasing weight (i.e., the parameter *WEIGHT*).
2	Rows are sorted in order of increasing number of nonzeros (i.e., the parameter *NSUBS*).
3	Rows are sorted in order of increasing minimum column subscripts (see below).
4	Rows are sorted in order of increasing maximum column subscripts (see below).
5	Rows are sorted in order of the parameter *ROWNUM* (see Section 2.2).

Here the *maximum* {*minimum*} column subscript of a row is the column subscript (index) of the *last* {*first*} nonzero in that row. Moreover, the column indices referred to those of the (column) *permuted* matrix obtained from *ORCOLB*.

The effectiveness of these strategies is not well understood, and varies with the problem. In the absence of any prior knowledge, setting *OPTION* to 4 is recommended. See [8,9,10] for discussions on the row ordering problem in the solution of sparse linear least squares problems.

Note:

The amount of storage required to perform row ordering may be substantially larger than those required by the other interface subroutines.

2.5. Module for numerical solution

The actual numerical computation which produces the solution β is initiated by executing the FORTRAN statement

CALL LSQSLV (TOL, TYPTOL, T)

where T is the working storage array.

The parameter *TOL* is a user-specified tolerance which is used to determine when a diagonal element of the upper triangular matrix produced in the numerical computation should be regarded as *numerically* zero. Suppose R denotes the upper triangular matrix. A diagonal element R_{ii} will be regarded as numerically zero if

$$| R_{ii} | \leq TOL \qquad \text{when} \quad TYPTOL = 0 \quad ,$$

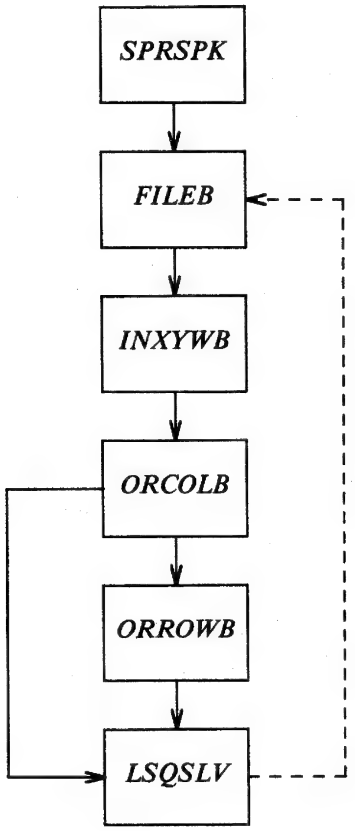
and

$$\frac{| R_{ii} |}{\max\limits_{k} | R_{kk} |} \leq TOL \qquad \text{when} \quad TYPTOL = 1 \quad .$$

The choice of *TOL* is usually problem-dependent. In general, *TOL* should be chosen so that it reflects the accuracy of X, Y, y and z. For example, if the input numerical values are known to be accurate to t significant digits, then one should use a relative test ($TYPTOL = 1$) and *TOL* should be set to about 10^{-t}.

3. Summary of the use of the basic interface modules

The following flowchart depicts the sequence of calls to SPARSPAK-B interface subroutines which occur when a constrained linear least squares problem is solved. After *LSQSLV* has been executed, the solution to the problem can be found in the first *NCOLS* locations in the working storage array *T*, where *NCOLS* is the third variable in the labelled common block *SPBUSR*.



A second and subsequent problems can be solved by simply starting at the beginning of the module sequence again, as implied by the broken line.

4. Save and restart facilities

As in SPARSPAK-A, SPARSPAK-B provides two subroutines called *SAVEB* and *RSTRTB* which allow the user to stop the calculation at some point, save the results on an external sequential file, and then restart the calculation at exactly the same point some time later. To save the results of the computation done thus far, the user executes the FORTRAN statement

CALL SAVEB (K, T)

where *K* is the FORTRAN logical unit on which the results are to be written, along with other information needed to restart the computation. If execution is then terminated, the state of the computation can be re-established by executing the FORTRAN statement

CALL RSTRTB (K, T)

Example 3 provided in Section 8 illustrates the use of *SAVEB* and *RSTRTB*.

Note that executing *SAVEB* does not destroy any information; the computation can proceed just as if *SAVEB* were not executed.

When errors occur in a module, the routines *SAVEB* and *RSTRTB* are useful in saving the results of previous modules executed successfully (see Section 6.3 and Example 4 in Section 8).

Another potential use of the *SAVEB* and *RSTRTB* modules is to make the working storage array *T* available to the user in the middle of a sparse matrix computation. After *SAVEB* has been executed, the working storage array *T* can be used by some other computation.

Finally, the *SAVEB* and *RSTRTB* modules allow the user to segment the computation into several distinct phases, and thereby reduce the amount of program that must be resident in storage at any given time.

Important Notes:

(a) In the subroutines *SAVEB* and *RSTRTB*, information is either written on or read from the FORTRAN logical unit *K* using *binary format.*

(b) If the subroutines *SAVEB* and *RSTRTB* are used, then before the user executes his program, he must define a file for the FORTRAN logical unit *K* using the appropriate system control statement or command. (This depends on the environment in which the program is being executed.) Furthermore, this file must be preserved by the user for later access by the *RSTRTB* subroutine.

(c) The external file for the logical unit *K* must be different from the working file specified in the *FILEB* statement.

5. Residual calculation

SPARSPAK-B provides a subroutine for computing the Euclidean norm of the residual vectors. The FORTRAN statement to be used is as follows:

CALL RESIDB (RESEQN, RESCON, T)

where T is the working storage array. After the subroutine is called,

$$RESEQN = || y - X\beta ||_2$$

and

$$RESCON = || z - Y\beta ||_2$$

where β is the computed solution.

Important Note:

RESIDB should be called only when the least squares solution has been computed (that is, after *LSQSLV* has been executed successfully).

6. Output from SPARSPAK-B

As noted earlier in Section 2.1, the user supplies a one-dimensional floating-point array *T*, from which all array storage is allocated. In particular, the interface allocates the first *NCOLS* storage locations in *T* for the solution vector of the constrained linear least squares problem. After all the interface modules for a particular problem have been successfully executed, the user can retrieve the solution from these *NCOLS* locations.

In addition to the least squares solution β, SPARSPAK-B may provide other information about the computation, depending upon the value of *MSGLVB* (the first variable in the common block *SPBUSR*), whether or not errors occur, and whether or not the module *STATSB* is called. This section discusses these features of SPARSPAK-B.

Notes:

Two logical output units (*IPRNTS* and *IPRNTE*) are required by SPARSPAK-B. Any information and/or statistics which the user has requested are recorded on the output unit *IPRNTS*, while any error messages that might be raised by SPARSPAK-B during the execution of the program are recorded on the output unit *IPRNTE*. These two logical output units are set in the initialization module *SPRSPK*, and it is the responsibility of the user and/or the computer installation to ensure that the files associated with these two logical units are defined before attempting to execute the program. The default output units are both 6. (These two output units are also used by SPARSPAK-A.)

6.1. Message level indicator (*MSGLVB*)

The first variable *MSGLVB* in the common block *SPBUSR* stands for "message level", and governs the amount of information printed by the interface modules. Its default value is two, and for this value a relatively small amount of summary information is printed, indicating the initiation of each phase. When *MSGLVB* is set to one by the user, only fatal error messages are printed; this option could be useful if SPARSPAK-B is being used in the "inner loop" of a large computation, where even summary information would generate excessive output. Increasing the value of *MSGLVB* (up to 4) provides increasingly detailed information about the computation. Note that the module *SPRSPK* sets *MSGLVB* to its default value; if the user wishes *MSGLVB* to be different from two, he must reset it *after SPRSPK* has been called.

In many circumstances, SPARSPAK-B will be embedded in still another "super package" which models phenomena producing sparse constrained linear least squares problems. Messages printed by SPARSPAK-B may be useless or even confusing to the ultimate users of the super package, or the super package may wish to field the error conditions and perhaps take some corrective action which makes the error messages irrelevant. Thus, *all* printing by SPARSPAK-B can be prevented by setting *MSGLVB* to zero.

We summarize our discussion in this section in the following table.

MSGLVB	amount of output
0	no information is provided.
1	only warnings and errors are printed.
2	warnings, errors and summary are printed.
3	warnings, errors, summary and some statistics are printed.
4	detailed information for debugging purposes.

Warning:

It should be noted that a high volume of output may be generated if *MSGLVB* is set to four, since the input data would also be echoed.

6.2. Statistics gathering (*STATSB*)

SPARSPAK-B gathers a number of statistics which the user will find useful if he is comparing SPARSPAK-B with other packages, or is going to solve numerous similar problems and wants to adjust the working storage to the minimum necessary. The package has a common block called *SPBDTA* containing variables whose values can be printed by executing the following FORTRAN statement.

CALL STATSB

The information printed includes

the size of the working storage array T,

the number of columns in X and Y,

the number of rows in A, B, E and F,

the number of nonzeros in the matrices X and Y,

the maximum number of nonzeros in the rows of X and Y,

the number of nonzeros in $\begin{pmatrix} E^T & A^T \end{pmatrix} \begin{pmatrix} E \\ A \end{pmatrix}$,

the number of off-diagonal nonzeros in the upper triangular matrix obtained after the numerical reduction,

the time used to find the column ordering,

the storage used by the column ordering subroutine,

the time used for data structure set-up,

the storage used by the storage allocation subroutine,

the time used to find the row ordering,

the storage used by the row ordering subroutine,

the time used for computing the solution,

the number of operations required by the solution subroutine,

the storage used by the solution subroutine,

the time used for computing the residuals,

the number of operations required by the residual calculation subroutine,

the storage used by the residual calculation subroutine,

the total time used by the solution process,

the maximum storage required by the solution process,

the Euclidean norm of the residual vectors.

Since the module *STATSB* can be called at any time, some of the above information may not be available, and will not be printed. Furthermore, the amount of information printed also depends on the message level *MSGLVB*. The word "operations" here means multiplicative operations (multiplications and divisions). Since most of the arithmetic performed in sparse matrix computation occurs in multiply-add pairs, the number of operations (as defined here) is a useful measure of the amount of arithmetic performed.

The reader is referred to the examples in Section 8 for more discussion about the output from *STATSB*.

6.3. Error messages (*IERRB*)

When a fatal error is detected, so that the computation cannot proceed, a positive code is assigned to *IERRB*. The user can simply check the value of *IERRB* to see if the execution of the module has been successful. This error flag can be used in conjunction with the save and restart feature described in Section 4 to retain the results of successfully completed parts of the computation, as shown by the program fragment below.

```
      .
      .
      .
      CALL   ORCOLB ( T )
      IF   ( IERRB .EQ. 0 )  GO TO 100
           CALL   SAVEB ( 3, T )
           STOP
100   CONTINUE
      .
      .
      .
```

The variable *IERRB* is set to the value $10 \times k + l$, where $1 \leq l \leq 9$ distinguishes the error, and k is determined by the type of module that sets *IERRB* positive.

k	interface modules
20	save and restart modules (*SAVEB* and *RSTRTB*)
21	problem initialization module (*FILEB*)
22	row input module (*INXYWB*)
23	column ordering and data structure set-up module (*ORCOLB*)
24	row reordering module (*ORROWB*)
25	solution module (*LSQSLV*)
26	residual calculation module (*RESIDB*)

6.3.1. Save and restart subroutines

IERRB	*SAVEB* and *RSTRTB*
201	Output unit given to *SAVEB* is not positive.
202	Input unit given to *RSTRTB* is not positive.
203	Insufficient storage in working storage array to restart the computational process. The minimum value of *MAXSB* required is printed in the error message.

6.3.2. Problem initialization subroutine

IERRB	*FILEB*
211	Input/output unit given to *FILEB* is not positive.

6.3.3. Row input subroutine

IERRB	*INXYWB*
221	Incorrect execution sequence. Probable cause of error: routine *FILEB* was not executed successfully.
222	Incorrect execution sequence. Probable cause of error: routine *ORCOLB* has already been executed. To start a new problem, *FILEB* must be called first.
223	Number of nonzeros (*NSUBS*) is not positive.
224	Input row type (*TYPE*) is invalid.
225	Input index (or subscript) is not positive.
226	Input weight (*WEIGHT*) is not positive.
227	Insufficient storage in working storage array to form matrix structure. The minimum value of *MAXSB* required is printed in the error message.

6.3.4. Column ordering subroutine

IERRB	*ORCOLB*
231	Incorrect execution sequence. Probable cause of error: routine *INXYWB* was not executed successfully.
232	Incorrect execution sequence. Probable cause of error: routine *ORCOLB* was called after having already been executed successfully.

IERRB	*ORCOLB*
233	Insufficient storage in working storage array to create adjacency structure. The minimum value of *MAXSB* required is printed in the error message. Response: execute *SAVEB*, and restart the computation using *ORCOLB* with *MAXSB* at least as large as that indicated in the error message.
234	Number of variables or columns (*NCOLS*) is zero.
235	Number of equations and constraints is zero (i.e., $m+p=0$).
236	Insufficient storage in working storage array to execute the column ordering routine. The minimum value of *MAXSB* required is printed in the error message. Response: execute *SAVEB*, and restart the computation using *ORCOLB* with *MAXSB* at least as large as that indicated in the error message.
237	Insufficient storage in working storage array to execute the storage allocation routine. The column ordering routine was successfully executed. Response: same as for error 236.
238	Insufficient storage in working storage array to hold the data structure pointers. The column ordering and storage allocation routines were successfully executed. Response: same as for error 236.

6.3.5. Row ordering subroutine

IERRB	*ORROWB*
241	Incorrect execution sequence. Probable cause of error: routine *ORCOLB* was not executed successfully.
242	Incorrect execution sequence. Probable cause of error: routine *ORROWB* was called after having already been executed successfully.
243	Input row ordering option (*OPTION*) is invalid.
244	Insufficient storage in working storage array to execute the row ordering routine. The minimum value of *MAXSB* required is printed in the error message. Response: execute *SAVEB*, and restart the computation using *ORROWB* with *MAXSB* at least as large as that indicated in the error message.

6.3.6. Solution subroutine

IERRB	*LSQSLV*
251	Incorrect execution sequence. Probable cause of error: routine *ORCOLB* or routine *ORROWB* was not executed successfully.
252	Incorrect execution sequence. Probable cause of error: routine *LSQSLV* was called after having already been executed successfully.
253	Input tolerance (*TOL*) is negative.
254	Input tolerance type (*TYPTOL*) is invalid.
255	Insufficient storage in working storage array to execute the solution routines. The minimum value of *MAXSB* required is printed in the error message. Response: execute *SAVEB*, and restart the computation using *LSQSLV* with *MAXSB* at least as large as that indicated in the error message.
256	routine *LSQSLV* fails to compute a singular value decomposition of intermediate small dense matrices.

6.3.7. Residual calculation subroutine

IERRB	*RESIDB*
261	Incorrect execution sequence. Probable cause of error: routine *LSQSLV* was not executed successfully.
262	Insufficient storage in working storage array to compute residuals. The minimum value of *MAXSB* required is printed in the error message. Response: execute *SAVEB*, and restart the computation using *RESIDB* with *MAXSB* at least as large as that indicated in the error message.

7. Summary listing of SPARSPAK-B interface subroutines

SPARSPAK initialization	*SPRSPK*
Problem initialization	*FILEB (IONUM)*
Row input	*INABWB (ROWNUM, TYPE, NSUBS, SUBS, VALUES, RHS, WEIGHT, T)*
Column ordering	*ORCOLB (T)*
Row ordering	*ORROWB (OPTION, T)*
Numerical solution	*LSQSLV (TOL, TYPTOL, T)*
Residual calculation	*RESIDB (RESEQN, RESCON, T)*
Print statistics	*STATSB*
Save Restart	*SAVEB (K, T)* *RSTRTB (K, T)*

8. Examples

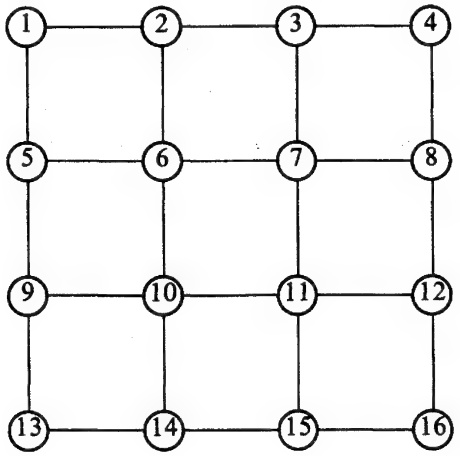
In this section, we provide several examples which illustrate how SPARSPAK-B can be used. The programs were compiled using the Berkeley f77 compiler and run on a DEC VAX 11/780 computer. A single precision version of SPARSPAK-A and SPARSPAK-B was used. All times reported are in seconds. It should be noted that the results may be different if a different version of SPARSPAK-A and SPARSPAK-B are used, or if the programs are compiled and run on a different computer.

The sample problems are variants of the following unconstrained linear least squares problem:

$$\min_{\beta} \| X\beta - y \|_2 \quad ,$$

where X is defined below.

Consider a $k \times k$ grid (see figure below, where $k=4$). There is a variable associated with each grid point, and for each square in the grid, there is a set of four equations involving the variables at the four grid points in that square. This gives rise to a sparse overdetermined system of linear equations and it will be solved in the least squares sense. The number of variables is $n=k^2$ and the number of equations is $m=4(k-1)^2$. For our purpose, the numerical values of the nonzeros in X and y are generated using a uniform random number generator.



Example 1

In this example, SPARSPAK-B is used to solve a constrained linear least squares problem

$$\min_{\beta \in \Omega} \| X\beta - y \|_2 \quad ,$$

where

$$\Omega = \{ \beta \mid \beta \text{ minimizes } \| Y\beta - z \|_2 \} \quad .$$

The matrix X is the same as the one defined on a $k \times k$ grid. The constraints are as follows.

$$Y = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \quad , \quad z = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$$

That is, the first and the second variables are assumed to have values 2 and 5 respectively.

The program begins by calling *SPRSPK* to initialize SPARSPAK-B, followed by a call to *FILEB* which tells SPARSPAK-B the unit number to be used to reference the temporary external file required by the package. Then the problem is generated and provided to SPARSPAK-B using the subroutine *INXYWB*. The column ordering is determined and data structure is set up by calling *ORCOLB*. Finally, the solution is obtained by executing *LSQSLV*. The subroutines *RESIDB* and *STATSB* are called to compute the norm of the residual vectors and to print out the statistics gathered by the package.

Note that in this example, the size of the working storage array T was 7500, while the statistic indicates that the maximum amount of storage required by any modules was only 514 which was the storage requirement in ORCOLB. Thus if problems with the same structure are to be solved, the user can change the size of T to 514.

Program

```
C--- SPARSPAK-B (ANSI FORTRAN) RELEASE III --- NAME = EX1                    1SPK
C  (C) UNIVERSITY OF WATERLOO   JANUARY 1984                                 2SPK
C***************************************************************************3SPK
C***************************************************************************4SPK
C**********     M A I N L I N E     P R O G R A M     **********             5SPK
C***************************************************************************6SPK
C***************************************************************************7SPK
C                                                                           8SPK
C     EXAMPLE 1 (SEE USER'S GUIDE).                                         9SPK
C                                                                          10SPK
C     REQUIREMENTS :                                                       11SPK
C        -- AN EXTERNAL FILE FOR UNIT 1.                                   12SPK
C        -- A RANDOM NUMBER GENERATOR (RANDOM).                            13SPK
C                                                                          14SPK
C***************************************************************************15SPK
C                                                                          16SPK
      INTEGER    SUBS(100)                                                 17SPK
      INTEGER    FILE  , I     , ICASE , IERRB , IPRNTE, IPRNTS,           18SPK
     1           ISEED , J     , K     , KGRID , KM1   , MAXINT,           19SPK
     1           MAXSB , MSGLVB, NCOLS , NDCONS, NDEQNS, NSCONS,           20SPK
     1           NSEQNS, NSUBS , OUTPUT, ROWNUM, TYPE  , TYPTOL            21SPK
      REAL       MCHEPS, RATIOL, RATIOS, TIME                              22SPK
      REAL              T(7500), VALUES(100)                               23SPK
      REAL              RESCON, RESEQN, RHS   , TOL   , WEIGHT             24SPK
C                                                                          25SPK
C***************************************************************************26SPK
```

```
C                                                                       27SPK
      COMMON /SPKSYS/ IPRNTE, IPRNTS, MAXINT, RATIOS, RATIOL,           28SPK
     1                MCHEPS, TIME                                     29SPK
      COMMON /SPBUSR/ MSGLVB, IERRB , MAXSB , NCOLS , NSEQNS,           30SPK
     1                NDEQNS, NSCONS, NDCONS                           31SPK
C                                                                       32SPK
C***********************************************************************33SPK
C                                                                       34SPK
C     ---------------                                                   35SPK
C     INITIALIZATION.                                                   36SPK
C     ---------------                                                   37SPK
      CALL  SPRSPK                                                      38SPK
C                                                                       39SPK
      FILE = 1                                                          40SPK
      OUTPUT = IPRNTS                                                   41SPK
      TOL = MCHEPS                                                      42SPK
      TOL = 100.0E0*TOL                                                 43SPK
      TYPTOL = 1                                                        44SPK
      ISEED = 1234567                                                   45SPK
C                                                                       46SPK
      MSGLVB = 2                                                        47SPK
      MAXSB = 7500                                                      48SPK
C                                                                       49SPK
      CALL  FILEB ( FILE )                                              50SPK
C                                                                       51SPK
C     --------------------------------                                  52SPK
C     GENERATE PROBLEM FROM THE GRID.                                   53SPK
C     --------------------------------                                  54SPK
      NSUBS = 4                                                         55SPK
      WEIGHT = 1.0E0                                                    56SPK
      TYPE = 1                                                          57SPK
C                                                                       58SPK
      ROWNUM = 0                                                        59SPK
C                                                                       60SPK
      KGRID = 5                                                         61SPK
      KM1 = KGRID - 1                                                   62SPK
      DO  400  I = 1, KM1                                               63SPK
          DO  300  J = 1, KM1                                           64SPK
C             -------------------                                       65SPK
C             GENERATE STRUCTURE.                                       66SPK
C             -------------------                                       67SPK
              SUBS(1) = (I - 1)*KGRID + J                               68SPK
              SUBS(2) = (I - 1)*KGRID + J + 1                           69SPK
              SUBS(3) = I*KGRID + J                                     70SPK
              SUBS(4) = I*KGRID + J + 1                                 71SPK
C             --------------------------------                          72SPK
C             GENERATE NUMERICAL VALUES USING                           73SPK
C             A RANDOM NUMBERGENERATOR.                                 74SPK
C             --------------------------------                          75SPK
              DO  200  ICASE = 1, 4                                     76SPK
                  DO  100  K = 1, NSUBS                                 77SPK
                      VALUES(K) = RANDOM(ISEED)                         78SPK
  100             CONTINUE                                              79SPK
                  ROWNUM = ROWNUM + 1                                   80SPK
                  RHS = RANDOM(ISEED)                                   81SPK
                  CALL  INXYWB ( ROWNUM, TYPE, NSUBS, SUBS,             82SPK
     1                           VALUES, RHS, WEIGHT, T )               83SPK
  200         CONTINUE                                                  84SPK
  300     CONTINUE                                                      85SPK
  400 CONTINUE                                                          86SPK
C                                                                       87SPK
C     ---------------------                                             88SPK
C     GENERATE CONSTRAINTS.                                             89SPK
C     ---------------------                                             90SPK
      ROWNUM = ROWNUM + 1                                               91SPK
      TYPE = 3                                                          92SPK
```

```
      NSUBS = 1                                                          93SPK
      SUBS(1) = 1                                                        94SPK
      VALUES(1) = 1.0E0                                                  95SPK
      RHS = 2.0E0                                                        96SPK
      WEIGHT = 1.0E0                                                     97SPK
      CALL   INXYWB ( ROWNUM, TYPE, NSUBS, SUBS, VALUES,                98SPK
     1                RHS, WEIGHT, T )                                  99SPK
C                                                                       100SPK
      ROWNUM = ROWNUM + 1                                               101SPK
      TYPE = 3                                                          102SPK
      NSUBS = 1                                                         103SPK
      SUBS(1) = 2                                                       104SPK
      VALUES(1) = 1.0E0                                                 105SPK
      RHS = 5.0E0                                                       106SPK
      WEIGHT = 1.0E0                                                    107SPK
      CALL   INXYWB ( ROWNUM, TYPE, NSUBS, SUBS, VALUES,                108SPK
     1                RHS, WEIGHT, T )                                  109SPK
C                                                                       110SPK
C     --------------                                                    111SPK
C     ORDER COLUMNS.                                                    112SPK
C     --------------                                                    113SPK
      CALL   ORCOLB ( T )                                               114SPK
C                                                                       115SPK
C     ---------------------------------------------                     116SPK
C     COMPUTE LEAST SQUARES SOLUTION AND RESIDUALS.                     117SPK
C     ---------------------------------------------                     118SPK
      CALL   LSQSLV ( TOL, TYPTOL, T )                                  119SPK
      CALL   RESIDB ( RESEQN, RESCON, T )                               120SPK
C                                                                       121SPK
C     ------------------------------------------                        122SPK
C     PRINT THE SOLUTION, FOUND IN THE FIRST NCOLS                      123SPK
C     LOCATIONS IN THE WORKING STORAGE ARRAY T.                         124SPK
C     ------------------------------------------                        125SPK
      WRITE (OUTPUT,11)   (T(K),K=1,NCOLS)                              126SPK
   11 FORMAT ( / 10H SOLUTION / (1P5E15.5) )                            127SPK
C     ---------------------------------------                           128SPK
C     PRINT STATISTICS GATHERED BY SPARSPAK-B.                          129SPK
C     ---------------------------------------                           130SPK
      CALL   STATSB                                                     131SPK
C                                                                       132SPK
      STOP                                                              133SPK
      END                                                               134SPK
```

Output

```
********** UNIVERSITY OF WATERLOO
********** SPARSE MATRIX PACKAGE
**********  ( S P A R S P A K )
**********       RELEASE   3
**********  (C) JANUARY 1984
********** ANSI FORTRAN
********** SINGLE PRECISION
********** LAST UPDATE JANUARY 1984


     OUTPUT UNIT FOR ERROR MESSAGES          6
     OUTPUT UNIT FOR STATISTICS              6

FILEB  - FILE INITIALIZATION ...
```

```
      INXYWB - INPUT ROWS ...

      ORCOLB - FIND COLUMN ORDERING ...

      LSQSLV - LEAST SQUARES SOLVE ...

      RESIDB - COMPUTE RESIDUAL ...

 SOLUTION
    2.00000e+00     5.00000e+00    -2.64674e+00     2.11532e+00    -3.95429e-01
   -3.00018e+00    -1.18907e+00     2.28644e-01     4.77276e-01    -6.61121e-03
    3.82073e-01     1.95622e+00     3.16640e-02     1.93660e-01     3.61554e-01
    5.77518e-01    -3.68422e-01     9.56243e-02     2.84502e-01     1.24321e+00
    6.70207e-01     5.10281e-01     3.93671e-01     1.24445e-02    -1.10439e+00

      STATSB - SYSTEM-B STATISTICS ...

           SIZE OF STORAGE ARRAY (MAXSB)       =      7500
           NUMBER OF COLUMNS (UNKNOWNS)        =        25
           NUMBER OF SPARSE EQUATIONS          =        64
           NUMBER OF DENSE EQUATIONS           =         0
           NUMBER OF SPARSE CONSTRAINTS        =         2
           NUMBER OF DENSE CONSTRAINTS         =         0
           TIME FOR COLUMN ORDERING            =     0.017
           STORAGE FOR COLUMN ORDERING         =     514.
           TIME FOR ALLOCATION                 =     0.033
           STORAGE FOR ALLOCATION              =     425.
           TIME FOR ROW ORDERING               =     0.
           STORAGE FOR ROW ORDERING            =     0.
           TIME FOR SOLUTION                   =     0.650
           OPERATION COUNT FOR SOLUTION        =    21758.
           STORAGE FOR SOLUTION                =     451.
           TIME FOR COMPUTING RESIDUAL         =     0.133
           OPN COUNT FOR COMPUTING RESIDUAL    =     390.000
           STORE FOR COMPUTING RESIDUAL        =     376.
           TOTAL TIME REQUIRED                 =     0.833
           MAXIMUM STORAGE REQUIRED            =     514.
           RESIDUAL IN EQUATIONS               =    7.900e+00
           RESIDUAL IN CONSTRAINTS             =    0.000e-01
```

Example 2

The purpose of this example is to illustrate the effect of dense rows. The problem being solved is the unconstrained linear least squares problem

$$\min_{\beta} \| X\beta - y \|_2 \quad ,$$

where X is partitioned into two portions

$$X = \begin{bmatrix} A \\ B \end{bmatrix} \quad .$$

Here A is the matrix defined on a $k \times k$ grid, and B and y_B are given below.

$$B = \begin{bmatrix} 1\,1\,1 \cdots 1\,1 \end{bmatrix} \quad , \quad y_B = \begin{bmatrix} 7 \end{bmatrix} \quad .$$

In the first run, all rows of X are treated as sparse ($TYPE = 1$). In the second run, the rows of A are regarded as sparse ($TYPE = 1$) whereas the row of B is treated as dense ($TYPE = 2$). Note the difference in storage requirements and execution times in the output.

Program 1

```
C--- SPARSPAK-B (ANSI FORTRAN) RELEASE III --- NAME = EX2A                    1SPK
C  (C)  UNIVERSITY OF WATERLOO   JANUARY 1984                                 2SPK
C*************************************************************************  3SPK
C*************************************************************************  4SPK
C**********     M A I N L I N E     P R O G R A M     **********              5SPK
C*************************************************************************  6SPK
C*************************************************************************  7SPK
C                                                                            8SPK
C     EXAMPLE 2 (SEE USER'S GUIDE).                                          9SPK
C                                                                           10SPK
C     REQUIREMENTS :                                                        11SPK
C        -- AN EXTERNAL FILE FOR UNIT 1.                                    12SPK
C        -- A RANDOM NUMBER GENERATOR (RANDOM).                             13SPK
C                                                                           14SPK
C*************************************************************************  15SPK
C                                                                           16SPK
      INTEGER    SUBS(100)                                                  17SPK
      INTEGER    FILE  , I     , ICASE , IERRB , IPRNTE, IPRNTS,            18SPK
     1           ISEED , J     , K     , KGRID , KM1   , MAXINT,            19SPK
     1           MAXSB , MSGLVB, NCOLS , NDCONS, NDEQNS, NSCONS,            20SPK
     1           NSEQNS, NSUBS , OUTPUT, ROWNUM, TYPE  , TYPTOL             21SPK
      REAL       MCHEPS, RATIOL, RATIOS, TIME                               22SPK
      REAL              T(7500), VALUES(100)                                23SPK
      REAL       RESCON, RESEQN, RHS   , TOL   , WEIGHT                     24SPK
C                                                                           25SPK
C*************************************************************************  26SPK
C                                                                           27SPK
      COMMON /SPKSYS/ IPRNTE, IPRNTS, MAXINT, RATIOS, RATIOL,               28SPK
     1                MCHEPS, TIME                                          29SPK
      COMMON /SPBUSR/ MSGLVB, IERRB , MAXSB , NCOLS , NSEQNS,               30SPK
     1                NDEQNS, NSCONS, NDCONS                                31SPK
C                                                                           32SPK
C*************************************************************************  33SPK
C                                                                           34SPK
C     ---------------                                                       35SPK
C     INITIALIZATION.                                                       36SPK
C     ---------------                                                       37SPK
```

```
      CALL  SPRSPK                                                    38SPK
C                                                                     39SPK
      FILE = 1                                                        40SPK
      OUTPUT = IPRNTS                                                 41SPK
      TOL = MCHEPS                                                    42SPK
      TOL = 100.0E0*TOL                                               43SPK
      TYPTOL = 1                                                      44SPK
      ISEED = 1234567                                                 45SPK
C                                                                     46SPK
      MSGLVB = 2                                                      47SPK
      MAXSB = 7500                                                    48SPK
C                                                                     49SPK
      CALL  FILEB ( FILE )                                            50SPK
C                                                                     51SPK
C     --------------------------------                                52SPK
C     GENERATE PROBLEM FROM THE GRID.                                 53SPK
C     --------------------------------                                54SPK
      NSUBS = 4                                                       55SPK
      WEIGHT = 1.0E0                                                  56SPK
      TYPE = 1                                                        57SPK
C                                                                     58SPK
      ROWNUM = 0                                                      59SPK
C                                                                     60SPK
      KGRID = 7                                                       61SPK
      KM1 = KGRID - 1                                                 62SPK
      DO  400  I = 1, KM1                                             63SPK
          DO  300  J = 1, KM1                                         64SPK
C                 -------------------                                 65SPK
C                 GENERATE STRUCTURE.                                 66SPK
C                 -------------------                                 67SPK
                  SUBS(1) = (I - 1)*KGRID + J                         68SPK
                  SUBS(2) = (I - 1)*KGRID + J + 1                     69SPK
                  SUBS(3) = I*KGRID + J                               70SPK
                  SUBS(4) = I*KGRID + J + 1                           71SPK
C                 --------------------------------                    72SPK
C                 GENERATE NUMERICAL VALUES USING                     73SPK
C                 A RANDOM NUMBER GENERATOR.                          74SPK
C                 --------------------------------                    75SPK
                  DO  200  ICASE = 1, 4                               76SPK
                      DO  100  K = 1, NSUBS                           77SPK
                          VALUES(K) = RANDOM(ISEED)                   78SPK
  100                 CONTINUE                                        79SPK
                      ROWNUM = ROWNUM + 1                             80SPK
                      RHS = RANDOM(ISEED)                             81SPK
                      CALL  INXYWB ( ROWNUM, TYPE, NSUBS, SUBS,       82SPK
     1                               VALUES, RHS, WEIGHT, T )         83SPK
  200             CONTINUE                                            84SPK
  300         CONTINUE                                                85SPK
  400 CONTINUE                                                        86SPK
C                                                                     87SPK
C     ------------------------                                        88SPK
C     GENERATE LAST EQUATION.                                         89SPK
C     ------------------------                                        90SPK
      ROWNUM = ROWNUM + 1                                             91SPK
      TYPE = 1                                                        92SPK
      NSUBS = KGRID*KGRID                                             93SPK
      DO  500  I = 1, NSUBS                                           94SPK
          SUBS(I) = I                                                 95SPK
          VALUES(I) = 1.0E0                                           96SPK
  500 CONTINUE                                                        97SPK
      RHS = 7.0E0                                                     98SPK
      WEIGHT = 1.0E0                                                  99SPK
      CALL  INXYWB ( ROWNUM, TYPE, NSUBS, SUBS, VALUES,              100SPK
     1               RHS, WEIGHT, T )                                 101SPK
C                                                                     102SPK
C     --------------                                                  103SPK
```

```
C       ORDER COLUMNS.                                                  104SPK
C       --------------                                                  105SPK
        CALL  ORCOLB ( T )                                              106SPK
C                                                                       107SPK
C       ------------------------------------------------                108SPK
C       COMPUTE LEAST SQUARES SOLUTION AND RESIDUALS.                   109SPK
C       ------------------------------------------------                110SPK
        CALL  LSQSLV ( TOL, TYPTOL, T )                                 111SPK
        CALL  RESIDB ( RESEQN, RESCON, T )                              112SPK
C                                                                       113SPK
C       ------------------------------------------------                114SPK
C       PRINT THE SOLUTION, FOUND IN THE FIRST NCOLS                    115SPK
C       LOCATIONS IN THE WORKING STORAGE ARRAY T.                       116SPK
C       ------------------------------------------------                117SPK
        WRITE (OUTPUT,11)  (T(K),K=1,NCOLS)                             118SPK
   11   FORMAT ( / 10H SOLUTION / (1P5E15.5) )                          119SPK
C       ---------------------------------------                         120SPK
C       PRINT STATISTICS GATHERED BY SPARSPAK-B.                        121SPK
C       ---------------------------------------                         122SPK
        CALL  STATSB                                                    123SPK
C                                                                       124SPK
        STOP                                                            125SPK
        END                                                             126SPK
```

Output

```
         ********** UNIVERSITY OF WATERLOO
         ********** SPARSE MATRIX PACKAGE
         **********  ( S P A R S P A K )
         **********      RELEASE   3
         **********  (C) JANUARY 1984
         ********** ANSI FORTRAN
         ********** SINGLE PRECISION
         ********** LAST UPDATE JANUARY 1984


              OUTPUT UNIT FOR ERROR MESSAGES          6
              OUTPUT UNIT FOR STATISTICS              6

         FILEB  - FILE INITIALIZATION ...

         INXYWB - INPUT ROWS ...

         ORCOLB - FIND COLUMN ORDERING ...

         LSQSLV - LEAST SQUARES SOLVE ...

         RESIDB - COMPUTE RESIDUAL ...

    SOLUTION
        4.27624e-01     3.45387e-02     1.26388e-01     3.57653e-01     3.13657e-01
        1.31271e-01    -6.67795e-01     3.41637e-02     4.47426e-01     4.08073e-01
        3.17152e-03     1.02795e-01     3.66493e-01     7.21198e-01    -3.65426e-01
        3.04172e-01    -9.93216e-02     5.65310e-01     4.02508e-01     5.26987e-02
       -6.21576e-02     2.51969e-01     5.16541e-01     1.59729e-01     6.74453e-02
       -3.53609e-02     4.96227e-01    -3.99947e-01    -3.46666e-01     3.04208e-01
        2.86251e-01     3.20507e-01     5.32274e-01     2.22485e-01     4.58984e-01
        6.90003e-01     4.31956e-01    -1.54123e-01     9.86527e-02     3.88556e-03
        1.82578e-01    -4.59611e-01    -1.64609e+00     7.08693e-01    -6.20304e-02
        5.53662e-01     1.02702e-03     7.23422e-01    -1.13267e-01
```

```
STATSB - SYSTEM-B STATISTICS ...

     SIZE OF STORAGE ARRAY (MAXSB)       =       7500
     NUMBER OF COLUMNS (UNKNOWNS)        =         49
     NUMBER OF SPARSE EQUATIONS          =        145
     NUMBER OF DENSE EQUATIONS           =          0
     NUMBER OF SPARSE CONSTRAINTS        =          0
     NUMBER OF DENSE CONSTRAINTS         =          0
     TIME FOR COLUMN ORDERING            =      0.050
     STORAGE FOR COLUMN ORDERING         =     5146.
     TIME FOR ALLOCATION                 =      0.200
     STORAGE FOR ALLOCATION              =     2795.
     TIME FOR ROW ORDERING               =        0.
     STORAGE FOR ROW ORDERING            =        0.
     TIME FOR SOLUTION                   =      2.550
     OPERATION COUNT FOR SOLUTION        =   248739.
     STORAGE FOR SOLUTION                =     1814.
     TIME FOR COMPUTING RESIDUAL         =      0.367
     OPN COUNT FOR COMPUTING RESIDUAL    =    915.000
     STORE FOR COMPUTING RESIDUAL        =     1667.
     TOTAL TIME REQUIRED                 =      3.167
     MAXIMUM STORAGE REQUIRED            =     5146.
     RESIDUAL IN EQUATIONS               =   3.663e+00
     RESIDUAL IN CONSTRAINTS             =   0.000e-01
```

Program 2

```
C--- SPARSPAK-B (ANSI FORTRAN) RELEASE III --- NAME = EX2B                  1SPK
C   (C)  UNIVERSITY OF WATERLOO   JANUARY 1984                             2SPK
C***********************************************************************  3SPK
C***********************************************************************  4SPK
C**********      M A I N L I N E     P R O G R A M      **********         5SPK
C***********************************************************************  6SPK
C***********************************************************************  7SPK
C                                                                          8SPK
C      EXAMPLE 2 (SEE USER'S GUIDE).                                       9SPK
C                                                                         10SPK
C      REQUIREMENTS :                                                     11SPK
C         -- AN EXTERNAL FILE FOR UNIT 1.                                 12SPK
C         -- A RANDOM NUMBER GENERATOR (RANDOM).                          13SPK
C                                                                         14SPK
C***********************************************************************  15SPK
C                                                                         16SPK
      INTEGER    SUBS(100)                                                17SPK
      INTEGER    FILE  , I     , ICASE , IERRB , IPRNTE, IPRNTS,          18SPK
     1           ISEED , J     , K     , KGRID , KM1   , MAXINT,          19SPK
     1           MAXSB , MSGLVB, NCOLS , NDCONS, NDEQNS, NSCONS,          20SPK
     1           NSEQNS, NSUBS , OUTPUT, ROWNUM, TYPE  , TYPTOL           21SPK
      REAL       MCHEPS, RATIOL, RATIOS, TIME                             22SPK
      REAL               T(7500), VALUES(100)                             23SPK
      REAL               RESCON, RESEQN, RHS   , TOL   , WEIGHT           24SPK
C                                                                         25SPK
C***********************************************************************  26SPK
C                                                                         27SPK
      COMMON /SPKSYS/ IPRNTE, IPRNTS, MAXINT, RATIOS, RATIOL,             28SPK
     1                MCHEPS, TIME                                        29SPK
      COMMON /SPBUSR/ MSGLVB, IERRB , MAXSB , NCOLS , NSEQNS,             30SPK
     1                NDEQNS, NSCONS, NDCONS                              31SPK
C                                                                         32SPK
C***********************************************************************  33SPK
```

```
C                                                                       34SPK
C     ---------------                                                   35SPK
C     INITIALIZATION.                                                   36SPK
C     ---------------                                                   37SPK
      CALL  SPRSPK                                                      38SPK
C                                                                       39SPK
      FILE = 1                                                          40SPK
      OUTPUT = IPRNTS                                                   41SPK
      TOL = MCHEPS                                                      42SPK
      TOL = 100.0E0*TOL                                                 43SPK
      TYPTOL = 1                                                        44SPK
      ISEED = 1234567                                                   45SPK
C                                                                       46SPK
      MSGLVB = 2                                                        47SPK
      MAXSB = 7500                                                      48SPK
C                                                                       49SPK
      CALL  FILEB ( FILE )                                              50SPK
C                                                                       51SPK
C     -------------------------------                                   52SPK
C     GENERATE PROBLEM FROM THE GRID.                                   53SPK
C     -------------------------------                                   54SPK
      NSUBS = 4                                                         55SPK
      WEIGHT = 1.0E0                                                    56SPK
      TYPE = 1                                                          57SPK
C                                                                       58SPK
      ROWNUM = 0                                                        59SPK
C                                                                       60SPK
      KGRID = 7                                                         61SPK
      KM1 = KGRID - 1                                                   62SPK
      DO  400  I = 1, KM1                                               63SPK
          DO  300  J = 1, KM1                                           64SPK
C             -------------------                                       65SPK
C             GENERATE STRUCTURE.                                       66SPK
C             -------------------                                       67SPK
              SUBS(1) = (I - 1)*KGRID + J                               68SPK
              SUBS(2) = (I - 1)*KGRID + J + 1                           69SPK
              SUBS(3) = I*KGRID + J                                     70SPK
              SUBS(4) = I*KGRID + J + 1                                 71SPK
C             -------------------------------                           72SPK
C             GENERATE NUMERICAL VALUES USING                           73SPK
C             A RANDOM NUMBER GENERATOR.                                74SPK
C             -------------------------------                           75SPK
              DO  200  ICASE = 1, 4                                     76SPK
                  DO  100  K = 1, NSUBS                                 77SPK
                      VALUES(K) = RANDOM(ISEED)                         78SPK
  100             CONTINUE                                              79SPK
                  ROWNUM = ROWNUM + 1                                   80SPK
                  RHS = RANDOM(ISEED)                                   81SPK
                  CALL  INXYWB ( ROWNUM, TYPE, NSUBS, SUBS,             82SPK
     1                           VALUES, RHS, WEIGHT, T )               83SPK
  200             CONTINUE                                              84SPK
  300         CONTINUE                                                  85SPK
  400 CONTINUE                                                          86SPK
C                                                                       87SPK
C     ------------------------------------------------                  88SPK
C     GENERATE LAST EQUATION AND TREAT IT AS DENSE.                     89SPK
C     ------------------------------------------------                  90SPK
      ROWNUM = ROWNUM + 1                                               91SPK
      TYPE = 2                                                          92SPK
      NSUBS = KGRID*KGRID                                               93SPK
      DO  500  I = 1, NSUBS                                             94SPK
          SUBS(I) = I                                                   95SPK
          VALUES(I) = 1.0E0                                             96SPK
  500 CONTINUE                                                          97SPK
      RHS = 7.0E0                                                       98SPK
      WEIGHT = 1.0E0                                                    99SPK
```

```
      CALL   INXYWB ( ROWNUM, TYPE, NSUBS, SUBS, VALUES,                100SPK
     1                RHS, WEIGHT, T )                                  101SPK
C                                                                        102SPK
C     --------------                                                     103SPK
C     ORDER COLUMNS.                                                     104SPK
C     --------------                                                     105SPK
      CALL   ORCOLB ( T )                                                106SPK
C                                                                        107SPK
C     ---------------------------------------------                      108SPK
C     COMPUTE LEAST SQUARES SOLUTION AND RESIDUALS.                      109SPK
C     ---------------------------------------------                      110SPK
      CALL   LSQSLV ( TOL, TYPTOL, T )                                   111SPK
      CALL   RESIDB ( RESEQN, RESCON, T )                                112SPK
C                                                                        113SPK
C     ---------------------------------------                            114SPK
C     PRINT THE SOLUTION, FOUND IN THE FIRST NCOLS                       115SPK
C     LOCATIONS IN THE WORKING STORAGE ARRAY T.                          116SPK
C     ---------------------------------------                            117SPK
      WRITE (OUTPUT,11)   (T(K),K=1,NCOLS)                               118SPK
   11 FORMAT ( / 10H SOLUTION / (1P5E15.5) )                             119SPK
C     ----------------------------------------                           120SPK
C     PRINT STATISTICS GATHERED BY SPARSPAK-B.                           121SPK
C     ----------------------------------------                           122SPK
      CALL   STATSB                                                      123SPK
C                                                                        124SPK
      STOP                                                               125SPK
      END                                                                126SPK
```

Output

```
********** UNIVERSITY OF WATERLOO
********** SPARSE MATRIX PACKAGE
**********  ( S P A R S P A K )
**********       RELEASE   3
**********  (C) JANUARY 1984
********** ANSI FORTRAN
********** SINGLE PRECISION
********** LAST UPDATE JANUARY 1984


     OUTPUT UNIT FOR ERROR MESSAGES          6
     OUTPUT UNIT FOR STATISTICS              6

FILEB  - FILE INITIALIZATION ...

INXYWB - INPUT ROWS ...

ORCOLB - FIND COLUMN ORDERING ...

LSQSLV - LEAST SQUARES SOLVE ...

RESIDB - COMPUTE RESIDUAL ...

 SOLUTION
   4.27621e-01    3.45408e-02    1.26385e-01    3.57654e-01    3.13655e-01
   1.31271e-01   -6.67797e-01    3.41631e-02    4.47427e-01    4.08074e-01
   3.17158e-03    1.02797e-01    3.66493e-01    7.21199e-01   -3.65426e-01
   3.04172e-01   -9.93229e-02    5.65309e-01    4.02507e-01    5.26985e-02
  -6.21579e-02    2.51969e-01    5.16541e-01    1.59730e-01    6.74474e-02
  -3.53588e-02    4.96227e-01   -3.99947e-01   -3.46664e-01    3.04207e-01
```

```
 2.86252e-01     3.20503e-01     5.32275e-01     2.22484e-01     4.58985e-01
 6.90001e-01     4.31955e-01    -1.54122e-01     9.86543e-02     3.88643e-03
 1.82577e-01    -4.59609e-01    -1.64609e+00     7.08693e-01    -6.20310e-02
 5.53660e-01     1.02709e-03     7.23423e-01    -1.13268e-01

 STATSB - SYSTEM-B STATISTICS ...

      SIZE OF STORAGE ARRAY (MAXSB)          =       7500
      NUMBER OF COLUMNS (UNKNOWNS)           =         49
      NUMBER OF SPARSE EQUATIONS             =        144
      NUMBER OF DENSE EQUATIONS              =          1
      NUMBER OF SPARSE CONSTRAINTS           =          0
      NUMBER OF DENSE CONSTRAINTS            =          0
      TIME FOR COLUMN ORDERING               =      0.117
      STORAGE FOR COLUMN ORDERING            =     1066.
      TIME FOR ALLOCATION                    =      0.017
      STORAGE FOR ALLOCATION                 =      902.
      TIME FOR ROW ORDERING                  =        0.
      STORAGE FOR ROW ORDERING               =        0.
      TIME FOR SOLUTION                      =      1.400
      OPERATION COUNT FOR SOLUTION           =   128442.
      STORAGE FOR SOLUTION                   =     1132.
      TIME FOR COMPUTING RESIDUAL            =      0.317
      OPN COUNT FOR COMPUTING RESIDUAL       =    915.000
      STORE FOR COMPUTING RESIDUAL           =      985.
      TOTAL TIME REQUIRED                    =      1.850
      MAXIMUM STORAGE REQUIRED               =     1132.
      RESIDUAL IN EQUATIONS                  =   3.663e+00
      RESIDUAL IN CONSTRAINTS                =   0.000e-01
```

Example 3

The effect of row ordering on execution time is illustrated. The problem being considered is the unconstrained problem defined on a $k \times k$ grid.

Note that when row ordering option (*OPTION*) is set to four, there is a reduction in execution time for *LSQSLV*. However, one should also note that the amount of storage required to perform row ordering is larger than that required by the other modules.

This example also illustrates the use of the save and restart facilities. After *ORCOLB* is called, *SAVEB* is executed to save the current state of the computation. After the problem is solved with no row ordering, the state after the execution of *ORCOLB* is restored by executing *RSTRTB* so that the problem can now be solved without invoking *FILEB*, *INXYWB*, and *ORCOLB* again.

Program

```
C--- SPARSPAK-B (ANSI FORTRAN) RELEASE III --- NAME = EX3                    1SPK
C   (C)  UNIVERSITY OF WATERLOO   JANUARY 1984                               2SPK
C*************************************************************************  3SPK
C*************************************************************************  4SPK
C**********      M A I N L I N E     P R O G R A M      **********           5SPK
C*************************************************************************  6SPK
C*************************************************************************  7SPK
C                                                                            8SPK
C      EXAMPLE 3 (SEE USER'S GUIDE).                                         9SPK
C                                                                           10SPK
C      REQUIREMENTS :                                                       11SPK
C          -- AN EXTERNAL FILE FOR UNIT 1.                                  12SPK
C          -- AN EXTERNAL FILE FOR UNIT 2.                                  13SPK
C          -- A RANDOM NUMBER GENERATOR (RANDOM).                           14SPK
C                                                                           15SPK
C*************************************************************************  16SPK
C                                                                           17SPK
       INTEGER    SUBS(100)                                                 18SPK
       INTEGER    FILE  , I     , ICASE , IERRB , IPRNTE, IPRNTS,           19SPK
      1           ISEED , J     , K     , KGRID , KM1   , MAXINT,           20SPK
      1           MAXSB , MSGLVB, NCOLS , NDCONS, NDEQNS, NSCONS,           21SPK
      1           NSEQNS, NSUBS , OPTION, ROWNUM, SAVE  , TYPE  ,           22SPK
      1           TYPTOL                                                    23SPK
       REAL       MCHEPS, RATIOL, RATIOS, TIME                              24SPK
       REAL              T(7000), VALUES(100)                               25SPK
       REAL              RESCON, RESEQN, RHS   , TOL   , WEIGHT             26SPK
C                                                                           27SPK
C*************************************************************************  28SPK
C                                                                           29SPK
       COMMON /SPKSYS/ IPRNTE, IPRNTS, MAXINT, RATIOS, RATIOL,              30SPK
      1                MCHEPS, TIME                                         31SPK
       COMMON /SPBUSR/ MSGLVB, IERRB , MAXSB , NCOLS , NSEQNS,              32SPK
      1                NDEQNS, NSCONS, NDCONS                               33SPK
C                                                                           34SPK
C*************************************************************************  35SPK
C                                                                           36SPK
C      ---------------                                                      37SPK
C      INITIALIZATION.                                                      38SPK
C      ---------------                                                      39SPK
       CALL  SPRSPK                                                         40SPK
C                                                                           41SPK
       FILE = 1                                                             42SPK
       SAVE = 2                                                             43SPK
```

```
      TOL = MCHEPS                                                      44SPK
      TOL = 100.0E0*TOL                                                 45SPK
      TYPTOL = 1                                                        46SPK
      ISEED = 1234567                                                   47SPK
C                                                                       48SPK
      MSGLVB = 2                                                        49SPK
      MAXSB = 7000                                                      50SPK
C                                                                       51SPK
      CALL  FILEB ( FILE )                                              52SPK
C                                                                       53SPK
C     -------------------------------                                   54SPK
C     GENERATE PROBLEM FROM THE GRID.                                   55SPK
C     -------------------------------                                   56SPK
      NSUBS = 4                                                         57SPK
      WEIGHT = 1.0E0                                                    58SPK
      TYPE = 1                                                          59SPK
C                                                                       60SPK
      ROWNUM = 0                                                        61SPK
C                                                                       62SPK
      KGRID = 10                                                        63SPK
      KM1 = KGRID - 1                                                   64SPK
      DO  400  I = 1, KM1                                               65SPK
          DO  300  J = 1, KM1                                           66SPK
C             -------------------                                       67SPK
C             GENERATE STRUCTURE.                                       68SPK
C             -------------------                                       69SPK
              SUBS(1) = (I - 1)*KGRID + J                               70SPK
              SUBS(2) = (I - 1)*KGRID + J + 1                           71SPK
              SUBS(3) = I*KGRID + J                                     72SPK
              SUBS(4) = I*KGRID + J + 1                                 73SPK
C             --------------------------------                          74SPK
C             GENERATE NUMERICAL VALUES USING                           75SPK
C             A RANDOM NUMBER GENERATOR.                                76SPK
C             --------------------------------                          77SPK
              DO  200  ICASE = 1, 4                                     78SPK
                  DO  100  K = 1, NSUBS                                 79SPK
                      VALUES(K) = RANDOM(ISEED)                         80SPK
  100             CONTINUE                                              81SPK
                  ROWNUM = ROWNUM + 1                                   82SPK
                  RHS = RANDOM(ISEED)                                   83SPK
                  CALL  INXYWB ( ROWNUM, TYPE, NSUBS, SUBS,             84SPK
     1                           VALUES, RHS, WEIGHT, T )               85SPK
  200             CONTINUE                                              86SPK
  300         CONTINUE                                                  87SPK
  400 CONTINUE                                                          88SPK
C                                                                       89SPK
C     --------------                                                    90SPK
C     ORDER COLUMNS.                                                    91SPK
C     --------------                                                    92SPK
      CALL  ORCOLB ( T )                                                93SPK
C                                                                       94SPK
C     --------------------------                                        95SPK
C     SAVE STATE OF COMPUTATION.                                        96SPK
C     --------------------------                                        97SPK
      CALL  SAVEB ( SAVE, T )                                           98SPK
C                                                                       99SPK
      OPTION = 0                                                       100SPK
  500 CONTINUE                                                         101SPK
          CALL  ORROWB ( OPTION, T )                                   102SPK
C                                                                      103SPK
C         ---------------------------------------------                104SPK
C         COMPUTE LEAST SQUARES SOLUTION AND RESIDUALS.                105SPK
C         ---------------------------------------------                106SPK
          CALL  LSQSLV ( TOL, TYPTOL, T )                              107SPK
          CALL  RESIDB ( RESEQN, RESCON, T )                           108SPK
C         ---------------------------------------                      109SPK
```

```
C          PRINT STATISTICS GATHERED BY SPARSPAK-B.                    110SPK
C          ----------------------------------------                    111SPK
           CALL  STATSB                                                112SPK
           IF  ( OPTION .EQ. 4 )  STOP                                 113SPK
C                                                                      114SPK
C          ------------------------------                              115SPK
C          RESTORE STATE OF COMPUTATION.                               116SPK
C          ------------------------------                              117SPK
           CALL  RSTRTB ( SAVE, T )                                    118SPK
           OPTION = 4                                                  119SPK
           GO TO 500                                                   120SPK
C                                                                      121SPK
       END                                                             122SPK
```

Output

```
********** UNIVERSITY OF WATERLOO
********** SPARSE MATRIX PACKAGE
**********  ( S P A R S P A K )
**********       RELEASE   3
**********  (C) JANUARY 1984
********** ANSI FORTRAN
********** SINGLE PRECISION
********** LAST UPDATE JANUARY 1984


      OUTPUT UNIT FOR ERROR MESSAGES          6
      OUTPUT UNIT FOR STATISTICS              6

FILEB  - FILE INITIALIZATION ...

INXYWB - INPUT ROWS ...

ORCOLB - FIND COLUMN ORDERING ...

SAVEB  - SAVE STORAGE VECTOR ...

ORROWB - FIND ROW ORDERING ...

LSQSLV - LEAST SQUARES SOLVE ...

RESIDB - COMPUTE RESIDUAL ...

STATSB - SYSTEM-B STATISTICS ...

      SIZE OF STORAGE ARRAY (MAXSB)     =      7000
      NUMBER OF COLUMNS (UNKNOWNS)      =       100
      NUMBER OF SPARSE EQUATIONS        =       324
      NUMBER OF DENSE EQUATIONS         =         0
      NUMBER OF SPARSE CONSTRAINTS      =         0
      NUMBER OF DENSE CONSTRAINTS       =         0
      TIME FOR COLUMN ORDERING          =     0.217
      STORAGE FOR COLUMN ORDERING       =    2269.
      TIME FOR ALLOCATION               =     0.050
      STORAGE FOR ALLOCATION            =    1966.
      TIME FOR ROW ORDERING             =       0.
      STORAGE FOR ROW ORDERING          =       0.
      TIME FOR SOLUTION                 =     4.683
      OPERATION COUNT FOR SOLUTION      =  379405.
      STORAGE FOR SOLUTION              =    2306.
```

```
      TIME FOR COMPUTING RESIDUAL        =        0.717
      OPN COUNT FOR COMPUTING RESIDUAL   =     1944.000
      STORE FOR COMPUTING RESIDUAL       =     2006.
      TOTAL TIME REQUIRED                =        5.667
      MAXIMUM STORAGE REQUIRED           =     2306.
      RESIDUAL IN EQUATIONS              =    4.823e+00
      RESIDUAL IN CONSTRAINTS            =    0.000e-01

RSTRTB - RESTART SYSTEM-B ...

ORROWB - FIND ROW ORDERING ...

LSQSLV - LEAST SQUARES SOLVE ...

RESIDB - COMPUTE RESIDUAL ...

STATSB - SYSTEM-B STATISTICS ...

      SIZE OF STORAGE ARRAY (MAXSB)      =      7000
      NUMBER OF COLUMNS (UNKNOWNS)       =       100
      NUMBER OF SPARSE EQUATIONS         =       324
      NUMBER OF DENSE EQUATIONS          =         0
      NUMBER OF SPARSE CONSTRAINTS       =         0
      NUMBER OF DENSE CONSTRAINTS        =         0
      TIME FOR COLUMN ORDERING           =        0.217
      STORAGE FOR COLUMN ORDERING        =     2269.
      TIME FOR ALLOCATION                =        0.050
      STORAGE FOR ALLOCATION             =     1966.
      TIME FOR ROW ORDERING              =        0.083
      STORAGE FOR ROW ORDERING           =     5850.
      TIME FOR SOLUTION                  =        3.983
      OPERATION COUNT FOR SOLUTION       =   325856.
      STORAGE FOR SOLUTION               =     2306.
      TIME FOR COMPUTING RESIDUAL        =        0.717
      OPN COUNT FOR COMPUTING RESIDUAL   =     1944.000
      STORE FOR COMPUTING RESIDUAL       =     2006.
      TOTAL TIME REQUIRED                =        5.050
      MAXIMUM STORAGE REQUIRED           =     5850.
      RESIDUAL IN EQUATIONS              =    4.823e+00
      RESIDUAL IN CONSTRAINTS            =    0.000e-01
```

Example 4

This example illustrates the use of the save and restart facilities to handle errors detected by SPARSPAK-B. The problem being solved is the same as the one in Example 3 and row ordering option is set to four. The size of the working storage array is initially set to 3000 which will be insufficient for *ORROWB* (as illustrated by Example 3). The state of the computation is saved by calling *SAVEB* when the error is detected after *ORROWB* is called. After adjusting the size of the working storage array, we then execute the second program. The routine *RSTRTB* is called to restore the state of the computation before *ORROWB* is called.

Program 1

```
C--- SPARSPAK-B (ANSI FORTRAN) RELEASE III --- NAME = EX4A                    1SPK
C  (C)  UNIVERSITY OF WATERLOO   JANUARY 1984                                 2SPK
C*************************************************************************  3SPK
C*************************************************************************  4SPK
C**********     M A I N L I N E     P R O G R A M     **********             5SPK
C*************************************************************************  6SPK
C*************************************************************************  7SPK
C                                                                            8SPK
C     EXAMPLE 4 (SEE USER'S GUIDE).                                          9SPK
C                                                                           10SPK
C     REQUIREMENTS :                                                        11SPK
C        -- AN EXTERNAL FILE FOR UNIT 1.                                    12SPK
C        -- AN EXTERNAL FILE FOR UNIT 2.                                    13SPK
C        -- A RANDOM NUMBER GENERATOR (RANDOM).                             14SPK
C                                                                           15SPK
C*************************************************************************  16SPK
C                                                                           17SPK
      INTEGER    SUBS(100)                                                  18SPK
      INTEGER    FILE  , I     , ICASE , IERRB , IPRNTE, IPRNTS,            19SPK
     1           ISEED , J     , K     , KGRID , KM1   , MAXINT,            20SPK
     1           MAXSB , MSGLVB, NCOLS , NDCONS, NDEQNS, NSCONS,            21SPK
     1           NSEQNS, NSUBS , OPTION, ROWNUM, SAVE  , TYPE  ,            22SPK
     1           TYPTOL                                                     23SPK
      REAL       MCHEPS, RATIOL, RATIOS, TIME                               24SPK
      REAL              T(3000), VALUES(100)                                25SPK
      REAL              RESCON, RESEQN, RHS   , TOL   , WEIGHT              26SPK
C                                                                           27SPK
C*************************************************************************  28SPK
C                                                                           29SPK
      COMMON /SPKSYS/ IPRNTE, IPRNTS, MAXINT, RATIOS, RATIOL,               30SPK
     1                MCHEPS, TIME                                         31SPK
      COMMON /SPBUSR/ MSGLVB, IERRB , MAXSB , NCOLS , NSEQNS,               32SPK
     1                NDEQNS, NSCONS, NDCONS                               33SPK
C                                                                           34SPK
C*************************************************************************  35SPK
C                                                                           36SPK
C     ---------------                                                       37SPK
C     INITIALIZATION.                                                       38SPK
C     ---------------                                                       39SPK
      CALL  SPRSPK                                                          40SPK
C                                                                           41SPK
      FILE = 1                                                              42SPK
      SAVE = 2                                                              43SPK
      OPTION = 4                                                            44SPK
      TOL = MCHEPS                                                          45SPK
      TOL = 100.0E0*TOL                                                     46SPK
      TYPTOL = 1                                                            47SPK
      ISEED = 1234567                                                       48SPK
```

```
C                                                                       49SPK
      MSGLVB = 2                                                        50SPK
      MAXSB = 3000                                                      51SPK
C                                                                       52SPK
      CALL  FILEB ( FILE )                                              53SPK
C                                                                       54SPK
C     ------------------------------                                    55SPK
C     GENERATE PROBLEM FROM THE GRID.                                   56SPK
C     ------------------------------                                    57SPK
      NSUBS = 4                                                         58SPK
      WEIGHT = 1.0E0                                                    59SPK
      TYPE = 1                                                          60SPK
C                                                                       61SPK
      ROWNUM = 0                                                        62SPK
C                                                                       63SPK
      KGRID = 10                                                        64SPK
      KM1 = KGRID - 1                                                   65SPK
      DO  400  I = 1, KM1                                               66SPK
          DO  300  J = 1, KM1                                           67SPK
C             ------------------                                        68SPK
C             GENERATE STRUCTURE.                                       69SPK
C             ------------------                                        70SPK
              SUBS(1) = (I - 1)*KGRID + J                               71SPK
              SUBS(2) = (I - 1)*KGRID + J + 1                           72SPK
              SUBS(3) = I*KGRID + J                                     73SPK
              SUBS(4) = I*KGRID + J + 1                                 74SPK
C             ------------------------------                            75SPK
C             GENERATE NUMERICAL VALUES USING                           76SPK
C             A RANDOM NUMBER GENERATOR.                                77SPK
C             ------------------------------                            78SPK
              DO  200  ICASE = 1, 4                                     79SPK
                  DO  100  K = 1, NSUBS                                 80SPK
                      VALUES(K) = RANDOM(ISEED)                         81SPK
  100             CONTINUE                                              82SPK
                  ROWNUM = ROWNUM + 1                                   83SPK
                  RHS = RANDOM(ISEED)                                   84SPK
                  CALL  INXYWB ( ROWNUM, TYPE, NSUBS, SUBS,             85SPK
     1                           VALUES, RHS, WEIGHT, T )               86SPK
  200         CONTINUE                                                  87SPK
  300     CONTINUE                                                      88SPK
  400 CONTINUE                                                          89SPK
      IF  ( IERRB .NE. 0 )  GO TO 500                                   90SPK
C                                                                       91SPK
C     ------------------------                                          92SPK
C     ORDER COLUMNS AND ROWS.                                           93SPK
C     ------------------------                                          94SPK
      CALL  ORCOLB ( T )                                                95SPK
      IF  ( IERRB .NE. 0 )  GO TO 500                                   96SPK
      CALL  ORROWB ( OPTION, T )                                        97SPK
      IF  ( IERRB .NE. 0 )  GO TO 500                                   98SPK
C                                                                       99SPK
C     ---------------------------------------------                    100SPK
C     COMPUTE LEAST SQUARES SOLUTION AND RESIDUALS.                    101SPK
C     ---------------------------------------------                    102SPK
      CALL  LSQSLV ( TOL, TYPTOL, T )                                  103SPK
      IF  ( IERRB .NE. 0 )  GO TO 500                                  104SPK
      CALL  RESIDB ( RESEQN, RESCON, T )                               105SPK
      IF  ( IERRB .NE. 0 )  GO TO 500                                  106SPK
C                                                                      107SPK
C     ---------------------------------------                          108SPK
C     PRINT STATISTICS GATHERED BY SPARSPAK-B.                         109SPK
C     ---------------------------------------                          110SPK
      CALL  STATSB                                                     111SPK
      STOP                                                             112SPK
C                                                                      113SPK
  500 CONTINUE                                                         114SPK
```

```
      CALL   SAVEB ( SAVE, T )                                    115SPK
      CALL   STATSB                                               116SPK
      STOP                                                        117SPK
C                                                                 118SPK
      END                                                         119SPK
```

Output

```
********** UNIVERSITY OF WATERLOO
********** SPARSE MATRIX PACKAGE
**********  ( S P A R S P A K )
**********       RELEASE   3
**********  (C) JANUARY 1984
********** ANSI FORTRAN
********** SINGLE PRECISION
********** LAST UPDATE JANUARY 1984


     OUTPUT UNIT FOR ERROR MESSAGES              6
     OUTPUT UNIT FOR STATISTICS                  6

FILEB  - FILE INITIALIZATION ...

INXYWB - INPUT ROWS ...

ORCOLB - FIND COLUMN ORDERING ...

ORROWB - FIND ROW ORDERING ...

EMSGB  - SYSTEM-B ERROR ...
     ORROWB - ERROR NUMBER                     244
     INSUFF. SPACE FOR ROW ORDERING,
     MAXSB MUST AT LEAST BE                   5850

SAVEB  - SAVE STORAGE VECTOR ...

STATSB - SYSTEM-B STATISTICS ...

     SIZE OF STORAGE ARRAY (MAXSB)     =      3000
     NUMBER OF COLUMNS (UNKNOWNS)      =       100
     NUMBER OF SPARSE EQUATIONS        =       324
     NUMBER OF DENSE EQUATIONS         =         0
     NUMBER OF SPARSE CONSTRAINTS      =         0
     NUMBER OF DENSE CONSTRAINTS       =         0
     TIME FOR COLUMN ORDERING          =     0.217
     STORAGE FOR COLUMN ORDERING       =    2269.
     TIME FOR ALLOCATION               =     0.050
     STORAGE FOR ALLOCATION            =    1966.
     TOTAL TIME REQUIRED               =     0.267
     MAXIMUM STORAGE REQUIRED          =    2269.
```

Program 2

```
C--- SPARSPAK-B (ANSI FORTRAN) RELEASE III --- NAME = EX4B                  1SPK
C  (C)  UNIVERSITY OF WATERLOO   JANUARY 1984                               2SPK
C*************************************************************************  3SPK
C*************************************************************************  4SPK
C**********     M A I N L I N E     P R O G R A M     **********             5SPK
C*************************************************************************  6SPK
C*************************************************************************  7SPK
C                                                                           8SPK
C     EXAMPLE 4 (SEE USER'S GUIDE).                                         9SPK
C                                                                          10SPK
C     REQUIREMENTS :                                                       11SPK
C        -- AN EXTERNAL FILE FOR UNIT 1.                                   12SPK
C        -- AN EXTERNAL FILE FOR UNIT 2.                                   13SPK
C        -- A RANDOM NUMBER GENERATOR (RANDOM).                            14SPK
C                                                                          15SPK
C*************************************************************************  16SPK
C                                                                          17SPK
      INTEGER    FILE   , IERRB , IPRNTE, IPRNTS, MAXINT, MAXSB ,          18SPK
     1           MSGLVB, NCOLS , NDCONS, NDEQNS, NSCONS, NSEQNS,          19SPK
     1           OPTION, SAVE  , TYPTOL                                   20SPK
      REAL       MCHEPS, RATIOL, RATIOS, TIME                             21SPK
      REAL              T(10000)                                          22SPK
      REAL              RESCON, RESEQN, TOL                               23SPK
C                                                                          24SPK
C*************************************************************************  25SPK
C                                                                          26SPK
      COMMON /SPKSYS/ IPRNTE, IPRNTS, MAXINT, RATIOS, RATIOL,              27SPK
     1                MCHEPS, TIME                                         28SPK
      COMMON /SPBUSR/ MSGLVB, IERRB , MAXSB , NCOLS , NSEQNS,              29SPK
     1                NDEQNS, NSCONS, NDCONS                               30SPK
C                                                                          31SPK
C*************************************************************************  32SPK
C                                                                          33SPK
C     ---------------                                                      34SPK
C     INITIALIZATION.                                                      35SPK
C     ---------------                                                      36SPK
      CALL  SPRSPK                                                         37SPK
C                                                                          38SPK
      FILE = 1                                                             39SPK
      SAVE = 2                                                             40SPK
      OPTION = 4                                                           41SPK
      TOL = MCHEPS                                                         42SPK
      TOL = 100.0E0*TOL                                                    43SPK
      TYPTOL = 1                                                           44SPK
C                                                                          45SPK
      MSGLVB = 2                                                           46SPK
      MAXSB = 10000                                                        47SPK
C                                                                          48SPK
      CALL  RSTRTB ( SAVE, T )                                             49SPK
C                                                                          50SPK
C     -----------                                                          51SPK
C     ORDER ROWS.                                                          52SPK
C     -----------                                                          53SPK
      CALL  ORROWB ( OPTION, T )                                           54SPK
C                                                                          55SPK
C     ---------------------------------------------                        56SPK
C     COMPUTE LEAST SQUARES SOLUTION AND RESIDUALS.                        57SPK
C     ---------------------------------------------                        58SPK
      CALL  LSQSLV ( TOL, TYPTOL, T )                                      59SPK
      CALL  RESIDB ( RESEQN, RESCON, T )                                   60SPK
C                                                                          61SPK
```

```
C     ----------------------------------------                    62SPK
C     PRINT STATISTICS GATHERED BY SPARSPAK-B.                    63SPK
C     ----------------------------------------                    64SPK
      CALL  STATSB                                                65SPK
      STOP                                                        66SPK
C                                                                 67SPK
      END                                                         68SPK
```

Output

```
********** UNIVERSITY OF WATERLOO
********** SPARSE MATRIX PACKAGE
**********  ( S P A R S P A K )
**********       RELEASE   3
**********  (C) JANUARY 1984
********** ANSI FORTRAN
********** SINGLE PRECISION
********** LAST UPDATE JANUARY 1984


      OUTPUT UNIT FOR ERROR MESSAGES          6
      OUTPUT UNIT FOR STATISTICS              6

RSTRTB - RESTART SYSTEM-B ...

ORROWB - FIND ROW ORDERING ...

LSQSLV - LEAST SQUARES SOLVE ...

RESIDB - COMPUTE RESIDUAL ...

STATSB - SYSTEM-B STATISTICS ...

      SIZE OF STORAGE ARRAY (MAXSB)      =      10000
      NUMBER OF COLUMNS (UNKNOWNS)       =        100
      NUMBER OF SPARSE EQUATIONS         =        324
      NUMBER OF DENSE EQUATIONS          =          0
      NUMBER OF SPARSE CONSTRAINTS       =          0
      NUMBER OF DENSE CONSTRAINTS        =          0
      TIME FOR COLUMN ORDERING           =      0.217
      STORAGE FOR COLUMN ORDERING        =     2269.
      TIME FOR ALLOCATION                =      0.050
      STORAGE FOR ALLOCATION             =     1966.
      TIME FOR ROW ORDERING              =      0.100
      STORAGE FOR ROW ORDERING           =     5850.
      TIME FOR SOLUTION                  =      3.933
      OPERATION COUNT FOR SOLUTION       =    324939.
      STORAGE FOR SOLUTION               =     2306.
      TIME FOR COMPUTING RESIDUAL        =      0.700
      OPN COUNT FOR COMPUTING RESIDUAL   =     1944.000
      STORE FOR COMPUTING RESIDUAL       =     2006.
      TOTAL TIME REQUIRED                =      5.000
      MAXIMUM STORAGE REQUIRED           =     5850.
      RESIDUAL IN EQUATIONS              =    4.823e+00
      RESIDUAL IN CONSTRAINTS            =    0.000e-01
```

---

9. References

[1] A. Bjorck, "A general updating algorithm for constrained linear least squares problems", SIAM J. Sci. Stat. Comput., 5 (1984), pp.394-402.

[2] E.C.H. Chu, J.A. George, J.W.H. Liu, and E.G.Y. Ng, "User's guide for SPARSPAK-A: A collection of modules for solving sparse systems of linear equations", (in preparation, 1984).

[3] J.A. George and M.T. Heath, "Solution of sparse linear least squares problems using Givens rotations", Linear Algebra and its Appl., 34 (1980), pp. 69-83.

[4] J.A. George, M.T. Heath, and E.G.Y. Ng, "A comparison of some methods for solving sparse linear least squares problems", SIAM J. Sci. Stat. Comput., 4 (1983), pp. 177-187.

[5] J.A. George, M.T. Heath, and E.G.Y. Ng, "Solution of sparse underdetermined systems of linear equations", (to appear in SIAM J. Sci. Stat. Comput., 1984).

[6] J.A. George and J.W.H. Liu, "The design of a user interface for a sparse matrix package", ACM Trans. on Math. Software, 5 (1979), pp. 134-162.

[7] J.A. George and J.W.H. Liu, Computer solution of large sparse positive definite systems, Prentice-Hall Inc., Englewood Cliffs, N.J. (1981).

[8] J.A. George, J.W.H. Liu, and E.G.Y. Ng, "Row ordering schemes for sparse Givens transformations, I. Bipartite graph model", (to appear in Linear Algebra and its Appl., 1984).

[9] J.A. George, J.W.H. Liu, and E.G.Y. Ng, "Row ordering schemes for sparse Givens transformations, II. Implicit graph model", (to appear in Linear Algebra and its Appl., 1984).

[10] J.A. George and E.G.Y. Ng, "On row and column orderings for sparse least squares problems", SIAM J. Numer. Anal., 20 (1983), pp. 326-344.

[11] J.A. George and E.G.Y. Ng, "On the design and implementation of SPARSPAK-B: Waterloo sparse constrained linear least squares package", (in preparation, 1984).

[12] M.T. Heath, "Some extensions of an algorithm for sparse linear least squares problems", SIAM J. Sci. Stat. Comput., 3 (1982), pp. 223-237.

[13] C.L. Lawson and R.J. Hanson, Solving least squares problems, Prentice-Hall Inc., Englewood Cliffs, N.J. (1974).

[14] J.W.H. Liu, "On multiple elimination in the minimum degree algorithm", Technical Report No. 83-03, Department of Computer Science, York University, Downsview, Ontario (1983).